Enums in Forth

Best Practices and Alternatives
Impromptu Talk

EuroForth'22 conference 2022-09

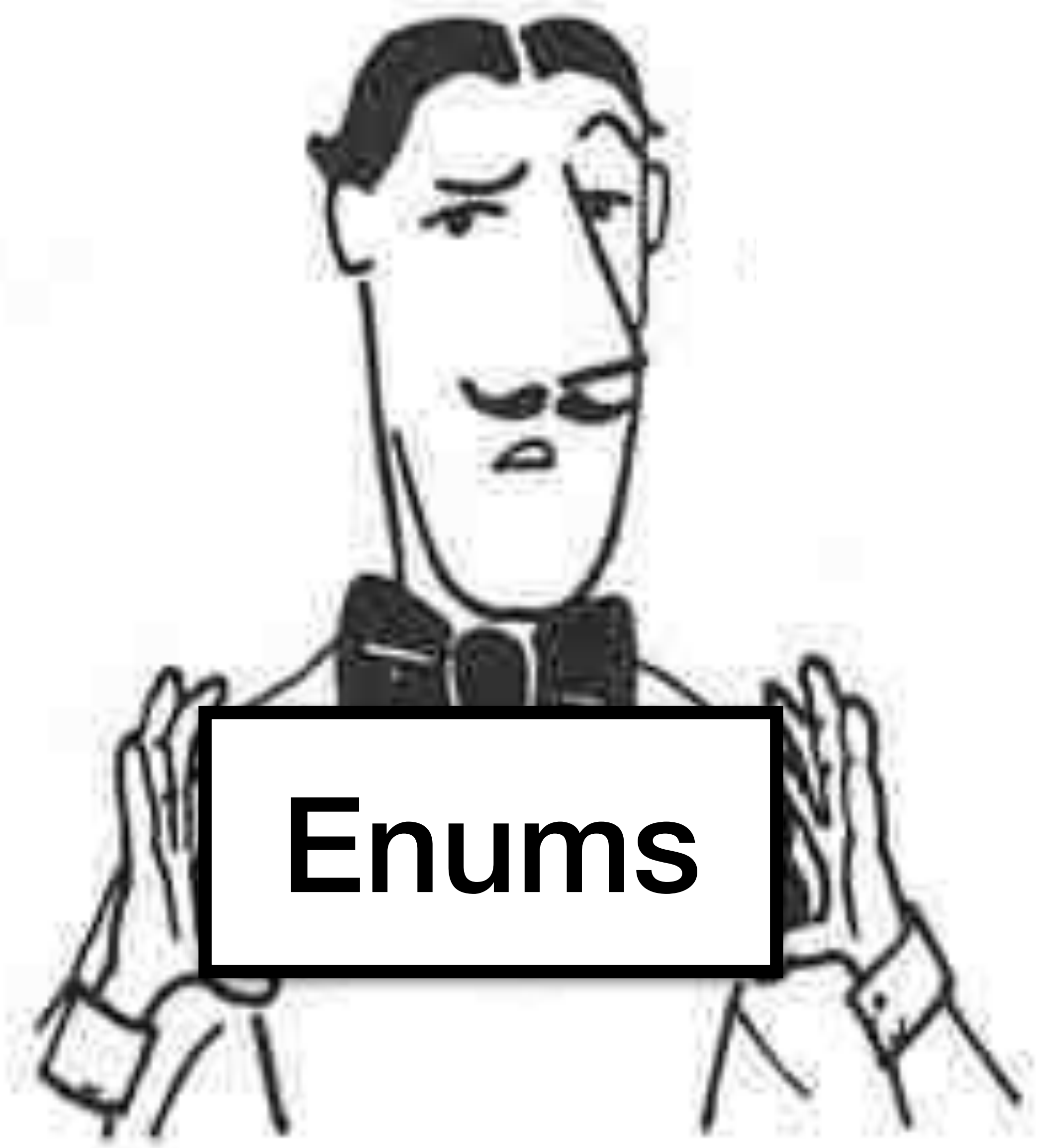
Ulrich Hoffmann

uho@ .de

Overview

Enums in Forth

- Enums in Forth
  - Explicit using *Forth Phrases*™
  - Nice Syntax - give names for phrases



Enums in Forth

- enumerations give names to values
- you don't remember all the numbers
- or they are likely to change
- or they are different on different systems

- use CONSTANTs

Nick Nelson:
"I don't like magic numbers."

Enums in Forth

- enumerations give Names to values - explicit with Constants

```
0 Constant black
1 Constant red
2 Constant green
3 Constant yellow

: .color ( c -- )
    dup black  = IF drop ." black"  EXIT THEN
    dup red    = IF drop ." red"    EXIT THEN
    dup green  = IF drop ." green"  EXIT THEN
    dup yellow = IF drop ." yellow" EXIT THEN
    ." color " . ;
```

Enums in Forth

- doing the calculations on your own

```
0
dup Constant black    1+ \ 0
dup Constant red      1+ \ 1
dup Constant green    1+ \ 2
dup Constant yellow   1+ \ 3
drop
```

- let the Forth interpreter do the calculation
- enum operations `dup` and `1+` are in different parts → combine them

```
dup Constant x    1+      →     dup 1+ swap Constant x
```

Enums in Forth

- doing the calculations on your own

```
0
dup 1+ swap Constant black    \ 0
dup 1+ swap Constant red      \ 1
dup 1+ swap Constant green    \ 2
dup 1+ swap Constant yellow   \ 3
drop
```

- explicit with Forth Phrases™
- a Forth Phrases is a sequence of inline forth words with no name
- Attention! Repeated phrases might be sign of bad factoring
- factorization given a name to phrases **over + swap -> bounds**

dup 1+ swap

Enums in Forth

- name the calculation - use the name to do the calculation implicitly
- traditionally (math) names this *iota* (greek letter **ι**)

```
: iota ( x -- x+1 x )  dup 1+ swap ;

0
iota Constant black     \ 0
iota Constant red       \ 1
iota Constant green     \ 2
iota Constant yellow    \ 3
drop
```

1 under+

Enums in Forth

- name the calculation - use the name to do the calculation implicitly
- traditionally (math) names this *iota* (greek letter **ι**)

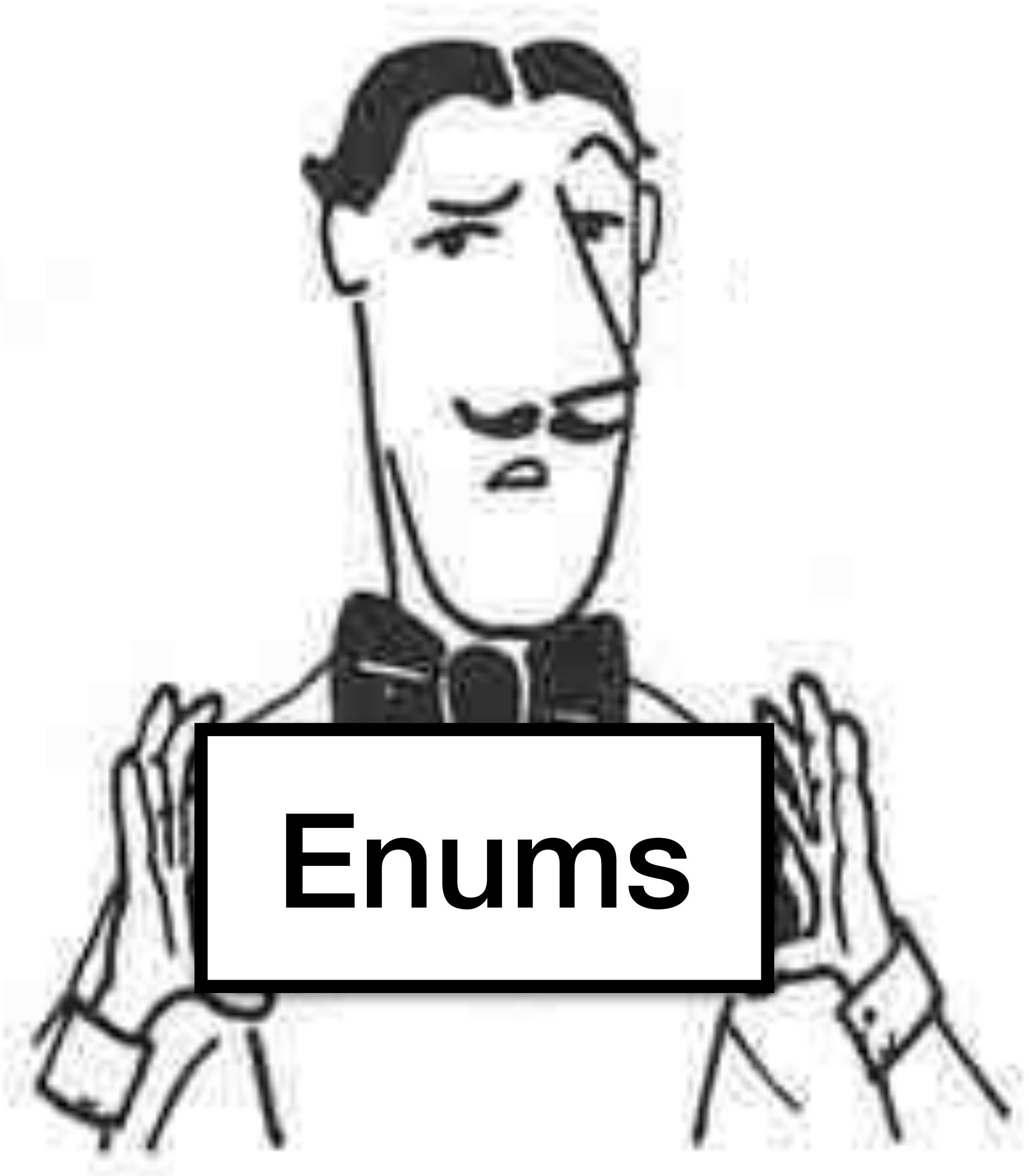
```
: ι ( x -- x+1 x )  dup 1+ swap ;

0
ι Constant black    \ 0
ι Constant red      \ 1
ι Constant green    \ 2
ι Constant yellow   \ 3
drop
```

1 under+

Enums in Forth

- Using a Defining word and capture Constant



```
: ι ( x -- x+1 x )  dup 1+ swap ;
: Enum ( n1 -- n2 )  ι Constant ;

0 Enum black    \ 0
  Enum red      \ 1
  Enum green    \ 2
  Enum yellow   \ 3
drop
```

```
: Enum ( n1 -- n2 )  dup Constant 1+ ;
```

see SwiftForth

```
enum{ black, red, green, yellow };
```

see Vfx